In the Specification:

Please amend paragraph 2 as follows:

[2] This application is related to U.S. Patent App. Ser. Nos. 10/684,102——— entitled IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-11-3), 10/684,053— entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-12-3); 10/683,929— entitled PIPELINE ACCELERATOR FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-13-3) and 10/683,932—— entitled PIPELINE ACCELERATOR HAVING MULTIPLE PIPELINE UNITS AND RELATED COMPUTING MACHINE AND METHOD (Attorney Docket No. 1934-15-3), which have a common filing date and owner and which are incorporated by reference.

Please amend paragraph 47 as follows:

[47] **FIG. 3** is a schematic block diagram of a computing machine ***40***, which has a peer-vector architecture according to an embodiment of the invention. In addition to a host processor ***42***, the peer-vector machine ***40*** includes a pipeline accelerator ***44***, which performs at least a portion of the data processing, and which thus effectively replaces the bank of coprocessors ***14*** in the computing machine ***10*** of **FIG. 1**. Therefore, the host-processor ***42*** and the accelerator ***44*** (or pipeline units thereof, as discussed below) are "peers" that can transfer data vectors back and forth. Because the accelerator ***44*** does not execute program instructions, it typically performs mathematically intensive operations on data significantly faster than a bank of coprocessors can for a given clock frequency. Consequently, by combining the decision-making ability of the processor ***42*** and the number-crunching ability of the accelerator ***44***, the machine ***40*** has the same abilities as, but can often process data faster than, a conventional computing machine such as the machine ***10***. Furthermore, as discussed below, providing the accelerator ***44*** with a communication interface that is compatible with the communication interface of the host processor ***42*** facilitates the design and modification of the machine ***40***, particularly where the processor's communication interface is an industry standard. And where the accelerator ***44*** includes one or more PLICs, the host processor ***42*** can hard configure

physical interconnectors within the accelerator by sending appropriate firmware to these PLICs. The host processor ***42*** may not only configure the accelerator ***44*** in this manner during initialization of the peer-vector machine ***40***, but it may have the ability to reconfigure the accelerator during operation of the peer-vector machine as discussed below and in previously cited U.S. Patent App. Serial No. 10/684,053—— entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-12-3). Moreover, the peer-vector machine ***40*** may also provide other advantages as described below and in the previously cited patent applications.

Please amend paragraph 49 as follows:

[49] The host processor ***42*** includes a processing unit ***62*** and a message handler ***64***, and the processor memory ***46*** includes a processing-unit memory ***66*** and a handler memory ***68***, which respectively serve as both program and working memories for the processor unit and the message handler. The processor memory ***46*** also includes an accelerator-configuration registry ***70*** and a message-configuration registry ***72***, which store firmware and configuration data that respectively allow the host processor ***42*** to configure the functioning of the accelerator ***44*** and the format of the messages that the message handler ***64*** sends and receives. The configuration of the accelerator ***44*** and the message handler ***64*** is further discussed in previously cited U.S. Patent App. Serial No. 10/684,053 —— entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-12-3), and the configuration of the accelerator ***44*** is also further discussed below in conjunction with **FIGS. 4-6**.

Please amend paragraph 51 as follows:

[51] The general operation of the peer-vector machine ***40*** is discussed in previously cited U.S. Patent App. Serial No. 10/684,102– entitled IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-11-3), the structure and operation of the host processor ***42*** is discussed in previously cited U.S. Patent App. Serial No. 10/684,053– entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD

(Attorney Docket No. 1934-12-3), and the structure and operation of the pipeline accelerator ***44*** is discussed in previously cited U.S. Patent App. Serial Nos. 10/683,929- entitled PIPELINE ACCELERATOR FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-13-3) and 10/683,932 ——— entitled PIPELINE ACCELERATOR HAVING MULTIPLE PIPELINE UNITS AND RELATED COMPUTING MACHINE AND METHOD (1934-15-3). The operating configurations of the PLICs that compose the accelerator ***44*** are discussed in previously cited U.S. Patent App. Serial No. 10/683,929— entitled PIPELINE ACCELERATOR FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-13-3) and below in conjunction with **FIGS. 4 – 6**.

Please amend paragraph 52 as follows:

[52] Referring to **FIGS. 4 – 6**, techniques for "hard" configuring the accelerator ***44*** PLICs are discussed. As alluded to above, the hard configuration of a PLIC is programmed by firmware and denotes the specific physical interconnections among the components of the PLIC, *i.e.*, how one logic block is electrically connected to another logic block. This is in contrast to the "soft" configuration, which denotes a higher-level configuration of an already-hard-configured PLIC. For example, a hard-configured PLIC may include a buffer, and may also include a register that allows one to soft configure the size of the buffer by loading corresponding soft-configuration data into the register. Soft configuration of the accelerator ***44*** is further discussed in previously cited U.S. Patent App. Serial Nos. 10/684,053— entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-12-3) and 10/683,929— entitled PIPELINE ACCELERATOR FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-13-3).

Please amend paragraph 53 as follows:

[53] **FIG. 4** is a block diagram of a pipeline unit ***78*** of the pipeline accelerator ***44*** of **FIG. 3** according to an embodiment of the invention. The hardwired pipelines $74_1 - 74_n$ (**FIG. 3**) are part of the pipeline unit ***78***, which, as discussed below, includes circuitry that,

e.g., controls the hardwired pipelines and allows them to receive, send, and store data. Although only one pipeline unit ***78*** is shown in **FIG.** 4, the accelerator ***44*** may include multiple pipeline units (each including at least some of the hardwired pipelines $74_1 - 74_n$) as discussed in previously cited U.S. Patent App. Serial No. 10/683,932— entitled PIPELINE ACCELERATOR HAVING MULTIPLE PIPELINE UNITS AND RELATED COMPUTING MACHINE AND METHOD (Attorney Docket No. 1934-15-3). As discussed below, in one implementation, the hard configuration of the pipeline unit ***78*** is programmable with firmware. This allows one to modify the functioning of the pipeline unit ***78*** by merely modifying the firmware. Furthermore, the host processor ***42*** (**FIG. 3**) can provide the modified firmware to the pipeline unit ***78*** during an initialization or reconfiguration of the peer-vector machine ***40*** (**FIG. 3**), and thus can eliminate the need for one to manually load the modified firmware into the pipeline unit.

Please amend paragraph 54 as follows:

[54] The pipeline unit ***78*** includes a pipeline circuit ***80***, such as a PLIC or an ASIC, the firmware memory ***52*** (where the pipeline circuit is a PLIC), and a data memory ***81***, which may all be disposed on a circuit board or card ***83***. The data memory ***81*** is further discussed in previously cited U.S. Patent App. Serial No. 10/683,929—— entitled PIPELINE ACCELERATOR FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-13-3)~~PROGRAMMABLE CIRCUIT AND RELATED COMPUTING MACHINE AND METHOD (Attorney Docket No. 1934-14-3)~~, and the combination of the pipeline circuit ***80*** and the firmware memory ***52*** forms a programmable-circuit unit.

Please amend paragraph 55 as follows:

[55] The pipeline circuit ***80*** includes a communication interface ***82***, which transfers data between a peer, such as the host processor ***42*** (**FIG. 3**), and the data memory ***81***, and also between the peer and the following other components of the pipeline circuit: the hardwired pipelines 74_1-74_n via a communication shell ***84***, a pipeline controller ***86***, an exception manager ***88***, and a configuration manager ***90***. The pipeline circuit ***80*** may also include an industry-standard bus interface ***91*** and a communication bus ***93***, which connects the

interface ***82*** to the interface ***91***. Alternatively, the functionality of the interface ***91*** may be included within the communication interface ***82*** and the bus ***93*** omitted. The structure and operation of the hardwired pipelines 74_1-74_n, controller ***86***, exception manager ***88***, configuration manager ***90***, and bus interface ***91*** are discussed in previously cited U.S. Patent App. Serial No. 10/683,929 entitled PIPELINE ACCELERATOR FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-13-3).

Please amend paragraph 58 as follows:

[58] The structure and operation of the communication interface ***82*** is further discussed in previously cited U.S. Patent App. Serial No. 10/683,929 entitled PIPELINE ACCELERATOR FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-13-3).

Please amend paragraph 71 as follows:

[71] When the peer-vector machine ***40*** is first powered on, the host processor ***42*** initializes itself as discussed in previously cited U.S. Patent App. Serial No. 10/684,053 entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-12-3), and the accelerator ***44*** partially initializes itself. More specifically, during this partial initialization, the pipeline circuit ***80*** downloads the initial-configuration firmware from the section ***114*** of the memory ***52***. As discussed above, in the initial configuration, the pipeline circuit ***80*** includes at least the communication interface ***82*** and test circuitry (not shown). After the pipeline circuit ***80*** is configured in the initial configuration, the test circuitry performs a self test of the pipeline circuit and the data memory ***81***, and provides the results of the self test to the host processor ***42*** via the test port ***96*** and the test bus ***63***. The firmware memory ***52*** may also perform a self test and provide the results to the host processor ***42*** via the test port ***104*** and the test bus ***63*** as discussed above in conjunction with **FIG. 5**.

Please amend paragraph 73 as follows:

[73] If an exception did occur, then the host processor ***42*** handles it in a predetermined manner. For example, if the host processor ***42*** does not receive a self-test result from the

pipeline circuit ***80***, then it may check, via the test bus ***63***, whether the initial-configuration firmware is stored in the section ***114*** of the firmware memory ***52***. If the initial-configuration firmware is not stored, then the host processor ***42*** may load the initial-configuration firmware into the section ***114*** via the pipeline bus ***50*** or the test bus ***63***, cause the pipeline circuit ***80*** to download this firmware, and then analyze the result of the self test. The host processor's handling of exceptions is further discussed in previously cited U.S. Patent App. Serial No. 10/684,053—— entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-12-3).

Please amend paragraph 83 as follows:

[83] Further details of the structure and operation of the pipeline unit ***124*** are discussed in previously cited U.S. Patent App. Serial No. 10/683,929— entitled PIPELINE ACCELERATOR FOR IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-13-3).

Please amend paragraph 89 as follows:

[89] When the peer-vector machine ***40*** is first powered on, the host processor ***42*** initializes itself as discussed in previously cited U.S. Patent App. Serial No. 10/684,053– entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-12-3), and the accelerator ***44*** partially initializes itself. More specifically, during this partial initialization, the pipeline circuits ***80a*** and ***80b*** download initial-configuration firmware from the sections ***114a*** and ***114b*** of the firmware memories ***52a*** and ***52b***, respectively. In the respective initial configurations, the pipeline circuit ***80a*** includes at least the communication interface ***82*** and test circuitry (not shown), and the pipeline circuit ***80b*** includes at least test circuitry (not shown). After the pipeline circuits ***80a*** and ***80b*** are configured in their respective initial configurations, the test circuit within each pipeline circuit performs a respective self test of the pipeline circuit — the test circuitry of one or both of the pipeline circuits ***80a*** and ***80b*** may also test the data memory ***81*** — and provides the results of these self tests to the host processor ***42*** via the test ports ***96a*** and ***96b***, respectively, and the test bus ***63***. The firmware memories ***52a*** and ***52b*** may also perform respective self tests and provide the

results to the host processor ***42*** via the test ports ***104a*** and ***104b***, respectively, and the test bus ***63*** as discussed above in conjunction with **FIG. 5**.

Please amend paragraph 91 as follows:

[91] If an exception did occur, then the host processor ***42*** handles it in a predetermined manner. For example, if the host processor ***42*** does not receive a self-test result from the pipeline circuit ***80a***, then it may check, via the test bus ***63***, whether the initial-configuration firmware is stored in the section ***114a*** of the firmware memory ***52a***. If the initial-configuration firmware is not stored, then the host processor ***42*** may load the initial-configuration firmware into the section ***114a***, cause the pipeline circuit ***80a*** to download this firmware, and then analyze the result of the self test. This example also applies to the pipeline circuit ***50b*** and the firmware memory ***52b***. The host processor's handling of exceptions is further discussed in previously cited U.S. Patent App. Serial No. 10/684,053— entitled COMPUTING MACHINE HAVING IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD (Attorney Docket No. 1934-12-3).